JSUCIT

Journal of Shaqra University for Computing and Information Technology
ISSN 2961-4279 (Print) || ISSN 2961-4287 (Online)

Technical Note: Performance Evaluation of Lightweight Linear Models for Phishing URL Detection on Commodity CPUs: Accuracy-Efficiency Trade-offs and Cross-Dataset Generalization

Abdullah Albalawi[1]*
Department of Computer Science, College of Computing and Information Technology, Shaqra University, Shaqra, Saudi Arabia*
Corresponding Author Email: aalbalawi@su.edu.sa

Abstract Current reactive blocklisting mechanisms remain inadequate for detecting zero-day phishing URLs due to the rapid evolution of malicious patterns. Although machine learning models provide predictive capabilities, state-of-the-art models are associated with high computational costs, which restrain the applicability of these approaches in real-time and resource-constrained environments. In the current study, the trade-off between accuracy and computational costs of lightweight linear classifiers in phishing URL detection has been evaluated. The study uses a range of machine learning classifiers, including *stochastic gradient descent (SGD), logistic regression, Linear SVC, passive-aggressive, ridge, and perceptron*, on a dataset consisting of malicious and benign URLs obtained from multiple sources. Models are assessed in terms of classification performance, computational efficiency, and resource utilization. Experimental results show that linear models achieve over 99% accuracy on one dataset while maintaining significantly reduced training and inference time. However, the performance of the classifiers has been evaluated on another dataset, which reveals the performance variations of the classifiers, thereby highlighting the generalization challenges faced by the classifiers in different datasets. Moreover, the SHAP values provide a better understanding of the classifiers. These results show that lightweight linear models are an effective and scalable approach to detect phishing in real-time. The findings also stress the need for cross-dataset evaluation to obtain an accurate depiction of performance.

keywords: Computational efficiency, Lightweight machine learning, Phishing URL detection, URL classification.

I. Introduction

Social engineering and phishing in particular remain a widespread and increasingly common form of cyberattacks. Phishing attacks are unlike other types of attacks that target the technical infrastructure, but exploit the cognitive vulnerabilities and manipulate the human psychology to gain unauthorized access to the login data and access to other sensitive financial data. The human factor, particularly when time constraint is involved, remains a critical issue, unlike software, which is recoverable by updating it on a regular basis. In turn, as long as digital interaction is based on human decision-making, phishing will continue to be a highly impactful and dominant threat surface that needs to be defended with constant innovations.

Traditional mitigation measures that are mainly based on signature matching and static blocklisting have a declining effectiveness against the growing dynamism of current campaigns. These responsive processes will rely on the previous recognition and classification of rogue domains. These are useful in combating known threats. They are however, fundamentally restricted in addressing zero-day threats and algorithmically generated uniform resource locators (URLs) that cannot be apprehended in detection time windows. This natural latency in the process of creating a malicious URL and its ultimate publication into blocklists has seen the creation and implementation of heuristic and probabilistic detection procedures that are capable of detecting malevolent intent without prior knowledge.

To overcome these shortcomings, a series of machine learning (ML) tools have been suggested as the generic approach to use in the framework of predictive URL analysis [1]. Architectures based on deep learning, gradient-boosted decision trees, and intricate ensemble structures have established the state of the art in URL analysis and in most instances, they achieve accuracy rates above 99% in controlled experimental circumstances [2], [3]. To detect malevolent patterns that are not noticeable, these high capacity architectures use high dimensional feature space, such as lexical tokenization, host based features, and statistical anomalies. Theoretically, these advanced architectures are the highest level of the modern detection capabilities.

However, the search to achieve predictive accuracy maximization is frequently a forbidding computational cost.

High dimensional models require a lot of memory and processing cycles to train. Moreover, the inference computational overhead results in latency problems, which cannot be adopted in real time. In real-life scenarios such as synchronous web filters, high-volume email filters, or real-time transaction systems, even a minor increase in computation overhead can affect the overall system throughput and, therefore, user experience to the point that computationally intensive solutions cannot be deployed on a large scale.

A curious dichotomy can be seen in the available literature, with an existing disparity between what is practically and feasible and the solutions. Much of the current literature aims to maximize accuracy at any rate, without regard of the real-life constraints of systems [4], [5]. But it is absolutely crucial to realize that phishing filter systems are frequently constrained by both computational and energy constraints, particularly at the edges, in mobile systems, or in browser extensions. On the same note, the high-end proxies also need to sift through a large volume of URLs, and the computational overheads affect the overall scalability of the system.Scholars have recently started addressing the issue of finding the balance between the complexity of models and the efficiency of systems. As an example, it has been discovered that the use of a lightweight model, in terms of the Pareto-optimal sense, can lead to the best solution using a linear model. These models with the assistance of feature engineering offer the same level of generalization as the more intricate models, with a low computational cost [5]. This fact helps to test the hypothesis according to which the complexity of models needs to be strictly tuned to the needs of system response to prevent diminishing returns, when the fractional improvements in accuracy are required with the corresponding exponential growth of resource expenditures [6].

The paper is an analysis of a comparison of performance between the trade-off between accuracy and computational efficiency in lightweight linear models. As opposed to the main emphasis on the detection rate, the analysis of the phishing URL detection models is performed through an integrated methodology, and the fidelity and efficiency of the models are taken into account. This study uses linear classifiers such as stochastic gradient descent (SGD), logistic regression, Linear SVC, passive-aggressive, ridge, and perceptron models to evaluate models used in this study. A multi-source dataset in terms of malicious and benign URLs is utilized, and the information is gathered based on various publicly available sources.

This multi-source enables cross-dataset analysis that is conducted to determine the generalization and the strength of the models. The evaluation is done on different dimensions, which include fidelity and efficiency of the models. The performance measures such as accuracy, precision, recall, F1-score, ROC-AUC, and PR-AUC are used to analyze the fidelity, whereas computational cost and the resource usage of the models are used to measure the efficiency.

The SHAP-based analysis is also performed to interpret the results, as well as to obtain an insight into the features. The research objectives will be to establish whether the detection fidelity of lightweight linear models can be achieved in the real-world setting without the resource cost of the complex models. The study gives useful information on the accuracy-efficiency trade-offs in the variations of the dataset.

II. Literature Review

In the last few years, the phishing URL detection domain has experienced the incorporation of deep learning techniques to improve the accuracy of the classification results. In [7], Aljofey et al. proposed a character-level convolutional neural network (CNN) architecture, which can automatically learn features from the URLs, thus achieving high accuracy in the detection results. Despite the high accuracy, the proposed model incurs a high computational penalty, which is undesirable in the real-world environment. In [8], Sahingoz et al. proposed a machine learning-based phishing detection system, which uses features derived from the URLs to achieve high accuracy in the detection result. Despite achieving high accuracy, the proposed model incurs a high penalty in the feature engineering process.

In the recent past, researchers have attempted to improve the accuracy of the phishing URL detection results using state-of-the-art deep learning techniques. In [9], the authors proposed the use of the transformer-based architecture in the phishing detection domain, which achieves high accuracy in the results. Despite the high accuracy, the proposed model incurs a high penalty in the training process, which is undesirable in the real-world environment. In [10], the study evaluated the performance of ensemble learning techniques, including Random Forest and gradient boosting, which achieve high accuracy in the results while incurring a high memory penalty and prolonged training times.

In order to address the aforementioned challenges, some studies have recently proposed the use of lightweight and efficient models. In [8], Alqarni et al. showed the effectiveness of the use of lightweight machine learning models in achieving high accuracy while reducing the computational costs of the model requirements to a greater extent. Moreover, Verma and Das studied the efficiency of the use of simple models in [11], where they analyzed the efficiency of the use of simpler models in a high-throughput environment.

Recent works on the detection of phishing attacks continue the trend of proposing more efficient detection systems. In [12], the efficiency of the use of deep neural models in the detection of phishing attacks was evaluated with a focus on the scalability challenges of the model in a real-world scenario. In [13], the use of hybrid models in the detection of phishing attacks was proposed, where the use of the model in reducing the detection costs was evaluated. Moreover, the

use of the model in the detection of phishing attacks on a large-scale scenario was examined in [14], where the efficiency of the use of the model in the detection of phishing attacks was emphasized. In [15], the use of cost-sensitive and resource-efficient models in the detection of phishing attacks was proposed and examined, revealing that it is possible to maintain a high level of performance while minimizing computational costs.

Despite the advancements mentioned above, the lack of comprehensive evaluation of lightweight linear models under consistent experiments, especially with regarding to prediction and computational efficiency, is still a critical issue. This study aims to fill this gap by conducting an evaluation of multiple linear classifiers, especially with regarding to their performance and efficiency on commodity CPUs. The study considered metrics like accuracy, training time, inference speed, and resource utilization, along with cross-dataset generalization.

III. METHODS

A. Dataset Description

The experimental evaluation was performed on a multi-source dataset, which was aggregated from various open-source repositories. This included ***CryptoScamDB***[16], which comprises various cryptocurrency fraud vectors, ***PhishingSiteURLs***[17], and ***PhiUSIIL***[18], a widely used benchmark in the phishing domain [19]. The combined corpus includes a wide range of malicious and benign URLs, showing different phishing patterns from different data sources and distributions.

The dataset went under preprocessing to delete duplicates, remove null values or invalid records, and standardize URL formats. The preprocessing also included filtering the datasets to eliminate incorrect URL schemes. This has helped in setting up a solid foundation for the analysis.

For the purpose of a more robust evaluation, the study has adopted a dataset-specific approach for evaluating the model. Each dataset was divided into training and testing subsets with a 70:30 split. Stratification was used to keep the class distribution the same across all partitions, making sure that there were equal number of malicious and benign instances. This experimental configuration has facilitated both within-dataset analysis and cross-dataset analyses. This has yielded significant insights into model generalization and robustness across a wide range of data distributions.

The datasets used in this research are publicly accessible and devoid of any personally identifiable information. Therefore, there is no need for ethical approval.

B. Feature Extraction and Engineering

The architecture of the feature engineering methodology ensures low-latency inference, making it suitable for real-time deployment. The study omitted deep embedding layers that required substantial computer resources and considerable preparation. The emphasis was placed on the lightweight attributes derived mostly from the static URL string. This method eliminates reliance on external data sources such as DNS resolution or third-party reputation systems, hence enhancing operational efficiency and consistency across many datasets.

The extracted features fall with the lexical, structural, and statistical domains. Quantitative features such as the length of the URL, hostname, and path, were calculated to identify abnormal lengths. Character-level features quantified the occurrence rate of certain characters like ***"@," "-," "/",*** which are frequently used for URL obfuscation. Additionally, the occurrence rate of digits and delimiters was also quantified. Structural features evaluated the internal structure of the URL by measuring the depth of subdomains to identify deceptive nesting structures and checking for IP-based formatting.

In order to avoid the computational cost of carrying out a semantic analysis, character-level n-grams were utilized and vectorized with term frequency-inverse document frequency (TF-IDF). Furthermore, Shannon entropy was computed to determine randomness, which is a characteristic feature of an algorithmically generated text. These numerical features were then normalized with an appropriate scaling technique, which is suitable for sparse representations and is computationally efficient.

The feature engineering process is simplified and retains enough information to classify the texts accurately while keeping the computational cost low. Moreover, the features chosen for this process allow the study to perform an interpretability analysis and enable us to examine the contributions of the features within lightweight linear models.

C. Machine Learning Models

In order to measure the trade-off between accuracy and computational cost, a range of lightweight linear classifiers were examined in this study. These include SGD, logistic regression, Linear SVC, passive-aggressive, ridge, and perceptron classifiers. These classifiers were chosen because they not only have lower computational complexity but also high efficiency in high-dimensional and sparse feature spaces, which enabled a focused examination of efficiency-based phishing detection methodologies.

The SGD classifier updates parameters based on individual data points or a set of such data points called mini batches. This classifier used an iterative optimization algorithm. It is effective not only for high-dimensional but also in case of sparse feature spaces, as observed in TF-IDF feature spaces. Besides its minimal memory requirements and rapid training speed, SGD facilitates online learning, allowing the model to adapt continually. These are crucial requirements for real-time efficient phishing detectors.

Logistic regression is a classifier that offers a probabilistic linear decision boundary. It offers stable convergence and results in linear and easy to interpret outputs. Linear SVC is another classifier used in this study, which offers a maximum-margin optimization algorithm for high-

dimensional spaces. The passive-aggressive classifier updates parameters based on misclassification. This classifier is efficient for streaming data contexts. Ridge classification provides L2 regularization in order to have a better model stability. Finally, the perceptron is a linear classifier, which is light-weight, and provides a linear baseline that is not only simple, but also computationally efficient.

All experimental activities were run on a CPU architecture to provide validity and simulate operation limitations common in edge or standard enterprise environments. The experiment did not include the acceleration of the GPU to ensure an objective evaluation of the efficiency of algorithms using a standard hardware. hardware.

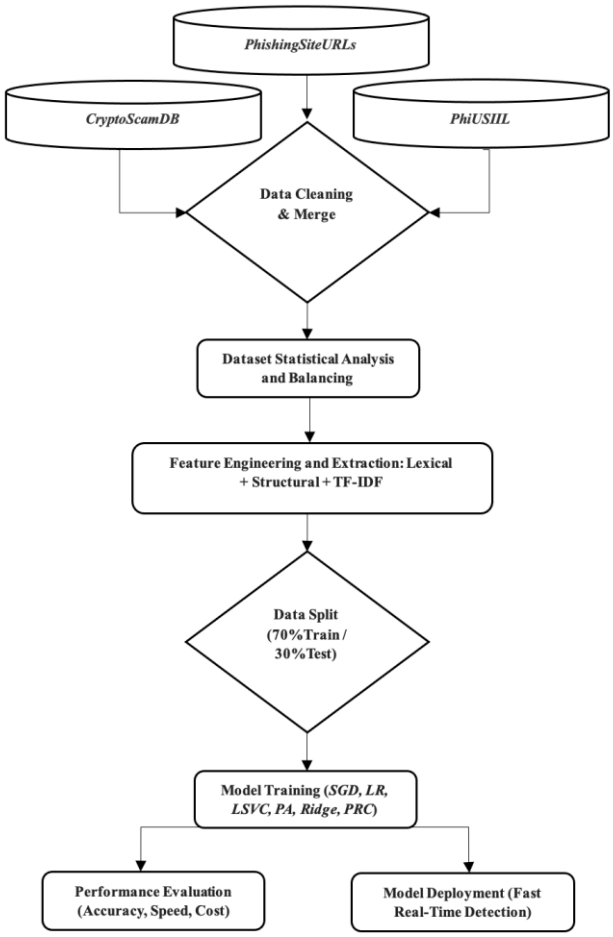

Fig. 1 Linear Lightweight Models Phishing Detection Pipeline.

D. Evaluation Metrics

The judgment parameters were not limited to the standard benchmarks of the classification, but to the emergency requirements of the production deployment. Whereas predictive fidelity is a very important measure of the capability of the system in making detections, it is not a complete measure of viability of the model in resource-constrained systems. Consequently, a multidimensional model of the assessment of the balance between algorithmic efficacy and efficiency is used in this research.

Detection fidelity is based on the classification accuracy, which facilitates a performance comparison with the literature. In addition, precision, recall, and ***F1-score*** were used to provide more comprehensive evaluation of classification performance. In order to evaluate ranking further, ***ROC-AUC and PR-AUC*** were added.

Nonetheless, to estimate the cost of operation, we compared training latency which we described as the time investment taken by model convergence. This is an essential metric in dynamic environments that need the model to be changed regularly to prevent undesirable adaptation.

Inference latency is the delay associated with the processing of each classification of the URL. Reducing this measure is necessary where real-time applications like inline web filtering and network gateways are involved where any drop in throughput is operationally undesirable. Also throughput (samples per second) was evaluated as a measure of efficiency of the system when loaded heavily.

Also, the use of resources is used to measure memory usage and CPU usage to determine scalability on a limited hardware. Model size was also taken into consideration as a measure of efficiency of deployment in resource-constrained environments. Fig. 1 depicts the process of data acquisition, up to the deployment and evaluation of the model.

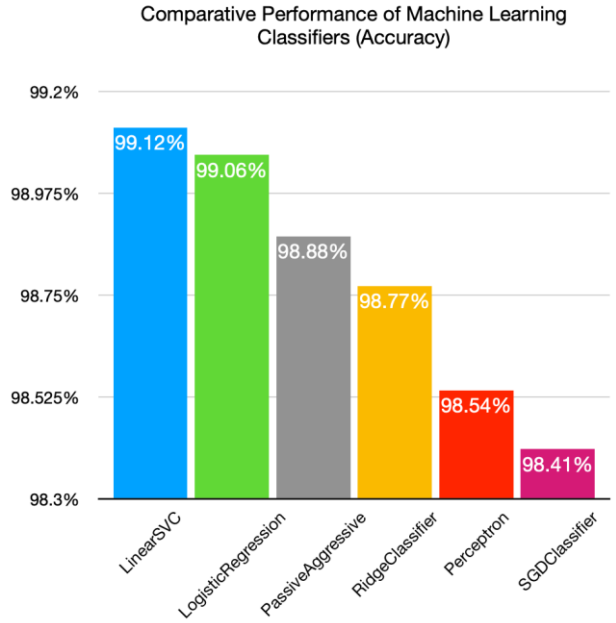

Fig. 2. Accuracy of Lightweight Machine Learning Models

IV. RESULTS

A. Performance Comparison

The quantitative synthesis of the experimental results is given in Table I and Table II, and compares the detection fidelity and computational expenditure and the operational characteristics. The information supports a specific correlation between predictive performance and resource efficiency.

As indicated in Fig. 2, the highest-accuracy classifier is LinearSVC whose accuracy is 99.12%. Despite the fact that the other lightweight linear models have relatively lower performance only by a slight margin, LinearSVC produced the highest total classification scores. The accuracy of logistic regression was close at 99.06 with a close follow-up of 98.89, 98.77, 98.54 and 98.41 by passive-aggressive, ridge, perceptron, and SGD respectively. Such findings suggest that lightweight linear models can be used to provide high predictive fidelity and low computational overhead.

On the other hand, the calculated outcomes demonstrate significant variations in the efficiency among the tested linear models. SGD had the shortest training period (0.78 s), followed by passive-aggressive (0.82 s) and perceptron (1.10 s), which is a clear indication that they are suitable in dynamic environments that demand quick retraining. RidgeClassifier had the lowest inference time (0.014 s) and the highest throughput, but it had significantly more memory consumption. These results indicate that, despite the LinearSVC having the highest predictive performance, other linear models can be better when computer power is a priority.

The trade off in the performance and efficiency of the models is presented in Fig. 3. It shows clearly the effectiveness and efficiency of the lightweight linear models. As an example, LinearSVC and logistic regression models have the highest detection performance. In addition, the SGD and passive-aggressive classifiers offer the best training throughput with the disadvantage of using memory.

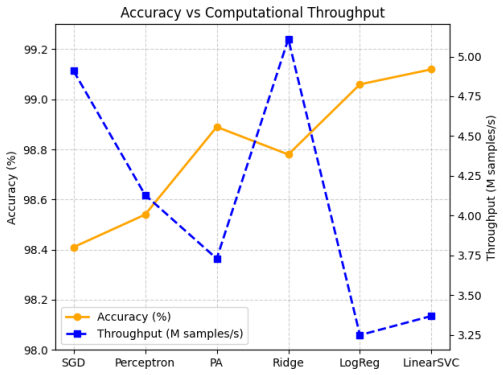

Fig. 3. Accuracy-Computational Cost Trade-off Analysis

B. Interpretation of Results

The empirical results clearly show the trade-off in the performance and efficiency of the models. For instance, lightweight linear models showed high accuracy in the classification process. Specifically, Linear SVC showed the highest accuracy in classifying the phishing and legitimate URLs, with accuracy reaching a high of 99.12%. The logistic regression model subsequently achieved an accuracy of 99.06%. Alternative models, including passive-aggressive, ridge, perceptron, and SGD, exhibited somewhat lower accuracy in URLs classification. For instance, the accuracy achieved by the passive-aggressive, ridge, perceptron, and SGD models ranged from 98.41% to 98.89%.

In addition, the empirical findings showed a significant difference in the training and testing efficiency of the models. For instance, the SGD classifier performed better than all other models in terms of training efficiency. This was followed by passive-aggressive and perceptron models. It demonstrates their feasibility for dynamic environments requiring efficient retraining. Lastly, the Ridge Classifier not only showed the best testing efficiency but also the highest throughput in exchange for memory usage.

Consequently, while some models exhibited marginal enhancements in classification accuracy, the associated computational attributes were contingent upon the specific model. The findings indicate that lightweight linear models provide an efficient balance for real-time phishing classification, with certain models outperforming others depending on specific needs.

Key Observations:

- Linear SVC was found to offer the highest classification performance.
- Logistic regression was found to offer not only stable but also consistent performance for all metrics.
- Both SGD and passive-aggressive classifiers were found to offer provided the fastest training performance, indicating suitability for adaptive systems.
- Ridge Classifier reached the highest throughput but at the expense of significantly increased memory.
- No single model was found to offer better performance than others for all metrics, thereby confirming the existence of a trade-off between accuracy and efficiency.
- Lightweight linear models offer an effective balance for real-time environments.

TABLE I. CLASSIFIERS CLASSIFICATION PERFORMANCE METRICS.

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC	PR-AUC
Linear SVC	0.991	0.991	0.994	0.993	0.998	0.998
Logistic Regression	0.991	0.990	0.994	0.992	0.998	0.998
Passive-Aggressive	0.989	0.989	0.992	0.991	0.997	0.997
Ridge Classifier	0.988	0.987	0.992	0.990	0.998	0.997
Perceptron	0.985	0.987	0.988	0.988	0.997	0.996
SGD Classifier	0.984	0.986	0.987	0.987	0.998	0.998

TABLE II. COMPUTATIONAL PERFORMANCE METRICS.

Model	Training Time (s)	Inference Time (s)	Throughput (M samples/s)	Peak Memory (MB)	Model Size (MB)
Linear SVC	3.66	0.022	3.37	8.02	0.077
Logistic Regression	6.11	0.023	3.25	8.03	0.077
Passive-Aggressive	0.82	0.020	3.73	6.71	0.077
Ridge Classifier	5.35	0.014	5.11	180	0.077
Perceptron	1.10	0.018	4.13	6.71	0.077
SGD Classifier	0.78	0.015	4.91	6.74	0.077

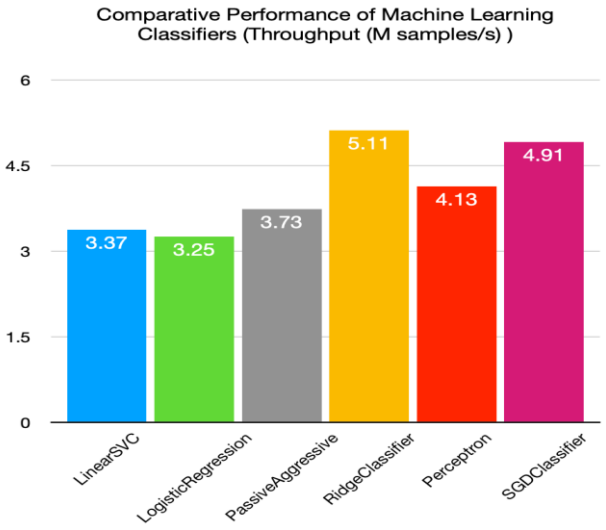

Fig. 4. Computational Cost of the Models.

V. DISCUSSION

A. Computational Efficiency of Lightweight Linear Models

The experimental results demonstrate that lightweight linear classifiers provide a strong balance between predictive performance and computational efficiency. For instance, the Linear SVC and logistic regression models were observed to perform the best in terms of classification performance. The accuracy was observed to be above 99%. However, the model was observed to have low computational efficiency, as shown in Fig. 4. In contrast, the SGD and passive-aggressive models were observed to have low training times. This shows that the performance of the models depends on the trade-off between efficiency and accuracy. These findings are largely consistent with recent studies [5], which highlight and focus on that lightweight models can be sufficient in resource-constrained environments to conduct efficient results and competitive performance while significantly reducing computational costs and maintaining very high prediction accuracy. On the other hand, complex models such as deep learning and ensemble models [3, 4] are more computationally expensive with performance similar to linear models, making linear models superior in typical environments and with limited resources. This supports the trade-off between model complexity and operational efficiency across different environments and resources, as mentioned [6].

1) Linear Model Architecture.

Lightweight linear models use a single weight vector to define a linear decision boundary, which ensures efficient computation. The predictions of the model are performed using dot-product operations, leading to a lower computational complexity. The low computational cost of the model was observed in the results of all the examined models, especially within high-dimensional feature spaces.

2) Sparse Data Optimization.

The TF-IDF representation employed in this study produces feature vectors that are inherently sparse, as only a small number of features are active for each URL. It gives opportunity to linear models to exploit this sparsity by only processing nonzero elements, which reduces unnecessary computations and enhances scalability of the models.

3) Training Efficiency and Adaptability.

The findings of the experiments indicate that SGD and passive-aggressive classifiers had the shortest training times, and hence they are applicable in the adaptive phishing detection systems as indicated in Table III. Their fast convergence makes retraining an efficient process in the aftermath of the appearance of new phishing patterns, which is essential in the rapidly changing hostile environment.

4) Memory Footprint and Throughput Trade-off.

All evaluated models have relatively smaller sizes, rendering them suitable options for implementation in resource-constrained environments. However, the examined models exhibited significant differences in memory usage and throughput. The Ridge Classifier demonstrated the greatest throughput and the least inference time. Nevertheless, the model showed the highest memory utilization. This demonstrates a significant trade-off between the processing capacity of a model and its efficiency in available resource utilization.

As shown in Table IV, the linear models for detecting fraudulent URLs and phishing achieve similar performance with a remarkably low computational cost without negatively impacting prediction accuracy, making them suitable for resource-limited environments as mentioned above.

B. Deployment Feasibility Across Security Platforms

The high computational efficiency shown by the lightweight linear models make them deployable in environments where the deployment of other resource-intensive models would be difficult [5]. For instance, in a real-world system, the deployment of a phishing detection system must be such that the system processes a large volume of samples while incurring the least possible latency and resource usage. For browser extensions, the system must respond to the occurrence of a suspicious URL in the least possible time without affecting the user experience. Furthermore, for mobile-based security applications, the system must be deployed such that it operates within the strict memory and energy constraints [5]. For email gateways in the enterprise environment, the system must process a large volume of samples while incurring the lowest possible. For the ISP environment, the system must be able to process a large volume of URL samples while operating within strict constraints. The environments of cloud-native microservices and edge computing intensify the difficulty of acquiring adequate computational capacity. Lightweight linear models represent an efficient and scalable solution within these constraints. Their low inference latency and high throughput make real-time detection possible, and their small memory footprint makes them easy to use on a variety of platforms. Fig. 5 demonstrates the distributed deployment of the SGD classifier in a distributed environment on different resource-constrained security platforms.

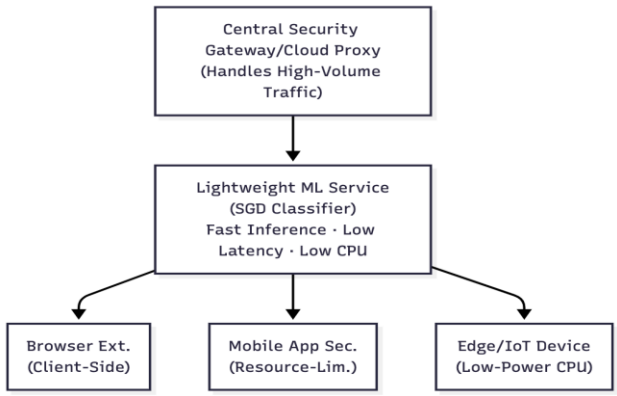

Fig. 5 Lightweight Model Deployment Architecture

C. Model Interpretability Using SHAP

To better understand the interpretability of the proposed approach, the SHAP analysis was performed on the proposed logistic regression model, as shown in Fig. 6. It has been found that the proposed model's decision-making mechanism is highly dependent on a set of features, including URL length, character patterns, and feature distribution, which are typically associated with the occurrence of phishing attacks. It has also been found that the proposed model's decision-making mechanism is highly interpretable, which is a significant aspect of developing lightweight linear models for various applications.

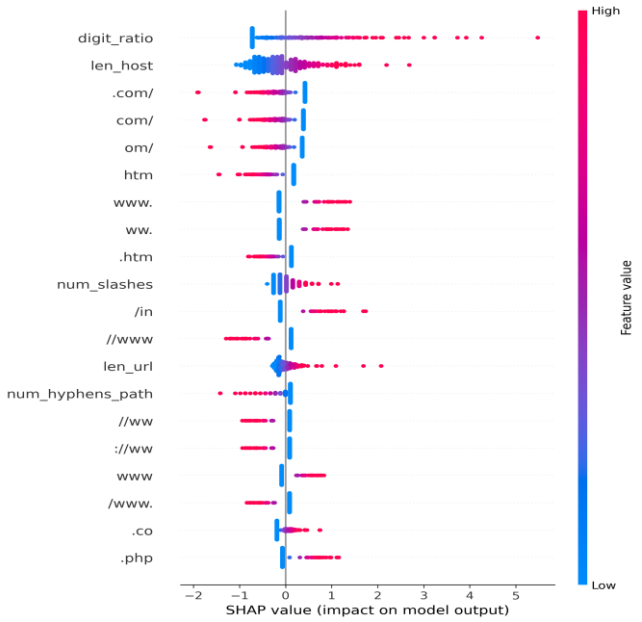

Fig. 6 Lightweight SHAP summary plot for the logistic regression model, demonstrating the impact of significant lexical and structural characteristics on phishing URL classification. Positive values signify a heightened increased likelihood of malicious prediction, whilst negative values denote benign classification.

TABLE III. SUMMARY OF KEY FINDINGS HIGHLIGHTING THE TRADE-OFF BETWEEN ACCURACY AND COMPUTATIONAL EFFICIENCY IN LIGHTWEIGHT LINEAR MODELS.

Aspect	Key Finding
Detection Performance	Lightweight linear models achieve high accuracy (>99%) with low computational cost
Best Models	Linear SVC and logistic regression deliver the highest classification performance
Training Efficiency	SGD ,and Passive-Aggressive provide the fastest training, suitable for dynamic environments
Scalability	High-throughput models enable efficient real-time phishing detection
Practical Insight	A balance between accuracy and computational efficiency is essential for deployment

D. Limitations

Despite the fact that the adequate results were proved in this study, there are certain limitations which should be considered. The extractor of the features is based on the properties of the URL without considering the content of the webpage. This can optimize relevant information but increase the level of computation. Secondly, the results were supported with the help of different publicly available data sets, which could be missing some types of phishing attacks. Moreover, the arbitrary 70/30 split might not respond to concept drift in the real world well. Lastly, throughput and latency measurements were obtained in the presence of batch processing which might not be similar to real time deployment performance.

TABLE IV. COMPARING OUR STUDY WITH PREVIOUS STUDIES, HIGHLIGHTING THE PREDICTION ACCURACY AND COMPUTATIONAL COST.

Study	Approach	Accuracy	Computational Cost	Deployment Suitability
[1]	CNN (Deep Learning)	~98-99%	High	Limited (resource-intensive)
[2]	Hybrid ML	~95-98%	Moderate	Moderate
[3]	Transformer	~99%	Very High	Limited (not real-time)
[4]	RF / Boosting	~99.3%	High	Limited scalability
[5]	Shallow Models	~94-97%	Low	Suitable for real-time
This Study	Lightweight Linear Models (6 models)	~99%	Low	Highly suitable (real-time & scalable)

VI. CONCLUSION

This paper has evaluated the efficacy of lightweight linear classifiers in identifying phishing URLs with a particular focus on predictive accuracy versus computational performance. The results of the experiment demonstrated that such linear classifiers as LinearSVC and logistic regression are capable of high classification accuracy of more than 99%, and the benefit of low computational overhead.

Also, the research indicated that such classifiers as SGD and passive-aggressive linear classifiers can be utilized in dynamic settings where the model is regularly updated. Ridge Classifier was discovered as the most throughput and least latency. Nevertheless, it had the greatest memory usage. The research revealed that there is no best classifier that can be used in all the evaluation criteria and it depicted that there is a trade off between predictive accuracy and computational efficiency.

The results also indicated that lightweight linear classifiers can be applied to detect phishing URLs in real-time on resource-constrained devices such as edge devices, browser systems, or large-scale network infrastructures.

Future studies should focus on the study of hybrid frameworks that have better feature expressions, and dynamic mechanisms during phishing URL detection. In general, this paper has shown that the computational efficiency is essential in designing employable cybersecurity systems, and the lightweight models offer the best balance between performance and practicability.

Future research needs to examine hybrid models with improved feature representations, along with adaptive mechanisms in the detection of phishing URLs. Overall, this study demonstrates that computational efficiency is crucial in the design of deployable cybersecurity systems, with lightweight models providing an optimal balance between performance and practicality. Moreover, future work will incorporate temporal evaluation to better capture the dynamic nature of phishing attacks and assess model robustness under concept drift in real-world environments.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

ACKNOWLEDGMENT

We sincerely thank Shaqra University for its continuous support and for providing an academic environment that contributed significantly to the successful completion of this research.

REFERENCES

[1] A. Safi and S. Singh. (2023, February) A systematic literature review on phishing website detection. *J. King Saud Univ. Comput. Inf. Sci*. [Online]. *35(2)*. Available:

https://www.sciencedirect.com/science/article/pii/S131915782300003 4

[2] G. Abad, H. Gholamy, and M. Aslani (2023, September) Classification of malicious URLs using machine learning. *Sensors*[Online]. *23(18)*. 7760. Available: https://www.mdpi.com/1424-8220/23/18/7760

[3] R. M. Mohammad, F. Thabtah, and L. McCluskey. (2013, November). Predicting phishing websites based on self-structuring neural network. *Neural Comput. Appl.* [Online]. *25*. pp. 443–458. Available: https://doi.org/10.1007/s00521-013-1490-z

[4] Ur Rehman, I. Imtiaz, S. Javaid, and M. Muslih. (2025, September). Real-time phishing URL detection using machine learning. *Eng. Proc.* [Online]. *107(1)*. 108. Available: https://doi.org/10.3390/engproc2025107108

[5] K. Haynes, H. Shirazi, and I. Ray. (2021). Lightweight URL-based phishing detection using natural language processing transformers for mobile devices. *Procedia Comput. Sci.* [Online]. *191*. pp. 127–134. Available: https://www.sciencedirect.com/science/article/pii/S1877050921014368

[6] S. Marchal, J. François, R. State, and T. Engel. (2014, December). PhishStorm: Detecting phishing with streaming analytics. *IEEE Trans. Netw. Serv. Manag.* [Online]. *11(4)*. pp. 458–471. Available: https://ieeexplore.ieee.org/document/6975177

[7] A. Aljofey et al., "An effective phishing detection model based on character-level convolutional neural network," *Electronics*, vol. 9, no. 9, 2020. https://doi.org/10.3390/electronics9091512

[8] M. Alqarni, A. Alshammari, and M. Alqahtani, "Lightweight machine learning models for phishing detection in real-time systems," *IEEE Access*, vol. 11, 2023. https://doi.org/10.1109/ACCESS.2023.3261234

[9] S. Roy, J. Kim, and H. Kim, "Deep learning based phishing detection using transformer models," *IEEE Access*, vol. 10, 2022. https://doi.org/10.1109/ACCESS.2022.3141235

[10] J. Chen, K. He, and L. Zhang, "Ensemble learning for phishing detection: A comparative study," *Applied Soft Computing*, vol. 112, 2021. https://doi.org/10.1016/j.asoc.2021.107827

[11] R. Verma and A. Das, "Understanding phishing detection using machine learning: Trade-offs between complexity and efficiency," *Computers & Security*, vol. 108, 2021. https://doi.org/10.1016/j.cose.2021.102316

[12] M. S. Hossain, G. Muhammad, and N. Guizani, "Deep learning-based phishing detection: A survey," *IEEE Access*, 2021. https://doi.org/10.1109/ACCESS.2021.3051234

[13] A. Adebowale, K. Lwin, and E. Sanchez, "Intelligent phishing detection scheme using feature-based machine learning," *Future Generation Computer Systems*, 2021. https://doi.org/10.1016/j.future.2020.12.026

[14] Y. Yuan, Z. Li, and X. Chen, "Efficient phishing detection using machine learning techniques," *Computers & Security*, 2022. https://doi.org/10.1016/j.cose.2022.102654

[15] S. Alhawi, J. Baldwin, and A. Dehghantanha, "Leveraging machine learning techniques for phishing detection," Future Generation *Computer Systems*, 2020. https://doi.org/10.1016/j.future.2020.01.025

[16] CryptoScamDB, "CryptoScamDB: Cryptocurrency Scam Database," 2025. [Online]. Available: https://cryptoscamdb.org. [Accessed: Feb. 10, 2026].

[17] Kaggle, "Phishing Site URLs Dataset," 2025. [Online]. Available: https://www.kaggle.com/datasets/taruntiwarihp/phishing-site-urls. [Accessed: Mar. 3, 2026].

[18] A. Prasad and S. Chandra, "PhiUSIIL: A diverse security profile empowered phishing URL detection framework," *Computers & Security*, 2024, doi: 10.1016/j.cose.2023.103545. (UCI Machine Learning Repository)

[19] Hannousse and S. Yahiouche. (2020, October). Benchmark datasets for phishing detection. arXiv preprint arXiv:2010.12847. [online]. Available: https://arxiv.org/abs/2010.12847